Amendments to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented method for a running parent process to collect exit information from a **defunct** child process~~, the child process~~ associated with ~~a nonterminated~~ **the running** parent process, the computer-implemented method comprising:

identifying a **running** parent process associated with ~~a~~ **the** defunct child process; and

modifying the **running** parent process associated with the defunct child process ~~wherein modifying the parent process~~ by ~~providing a thread associated with~~ **creating an agent thread inside** the **running** parent process**, the agent thread being operable to force the running parent process to collect** ~~that collects~~ exit information ~~associated with~~ **for** the defunct child process**, thereby enabling the collection of exit information for the defunct child process associated with the running parent process without terminating the running parent process**.

2. (Canceled)

3. (Canceled)

4. (Original) The computer-implemented method of claim 1, wherein modifying the parent process comprises altering the parent process to invoke wait() or waitpid().

5. (Original) The computer-implemented method of claim 1, wherein a control criterion is used to determine whether to modify the parent process.

6. (Original) The computer-implemented method of claim 5, wherein the control criterion comprises determining whether the parent process is stopped.

7. (Original) The computer-implemented method of claim 6, wherein the control criterion comprises determining whether the defunct process is a newly defunct process, wherein the parent is modified only if it is determined that the defunct process is not a newly defunct process.

8. (Original) The computer-implemented method of claim 6, wherein the control criterion comprises determining that the parent process is not an initial process.

9. (Currently Amended) A computer-implemented method for reaping a defunct child process associated with a **running** parent process, the computer-implemented method comprising:

identifying a defunct child process **associated with the running parent process;**

attaching an agent thread to ~~a~~ **the running** parent process to allow modification of the **running** parent process, wherein the **running** parent process is modified to reap the defunct child process **without terminating the running parent process**.

10. (Original) The computer-implemented method of claim 9, wherein a control criterion is used to determine whether to modify the parent process.

11. (Original) The computer-implemented method of claim 10, wherein the control criterion comprises determining whether the parent process is stopped.

12. (Original) The computer-implemented method of claim 10, wherein the control criterion comprises determining whether the defunct process is a newly defunct process, wherein the parent is modified only if it is determined that the defunct process is not a newly defunct process.

13. (Original) The computer-implemented method of claim 10, wherein the control criterion comprises determining that the parent process is not an initial process.

14. (Currently Amended) ~~The~~ **A** computer program product**, stored on a machine-readable medium,** ~~comprising computer code~~ for a running parent process to collect exit information from a **defunct** child process~~, the child process~~ associated with ~~a~~

~~nonterminated~~ **the running** parent process, the computer program product comprising **instructions operable to cause a computer to**:

~~computer code for~~ identify~~ing~~ a **running** parent process associated with ~~a~~**the** defunct child process; and

~~computer code for~~ modify~~ing~~ the **running** parent process associated with the defunct child process **by creating an agent thread inside, wherein** ~~modifying~~ the **running** parent process**, the agent thread being operable to force** ~~allows~~ the **running** parent process to collect exit information ~~associated with~~ **for** the defunct child process**, thereby enabling the collection of exit information for the defunct child process associated with the running parent process without terminating the running parent process.**

15. (Canceled)

16. (Canceled)

17. (Currently Amended) An apparatus for a running parent process to collect exit information from a **defunct** child process~~, the child process~~ associated with ~~a~~ ~~nonterminated~~ **the running** parent process, the apparatus comprising:

means for identifying a **running** parent process associated with ~~a~~ **the** defunct child process; and

means for modifying the **running** parent process associated with the defunct child process ~~wherein modifying the parent process~~ by ~~providing a thread associated with~~ **creating an agent thread inside** the **running** parent process**, the agent thread being operable to force the running parent process to collect** ~~that collects~~ exit information ~~associated with~~ **for** the defunct child process**, thereby enabling the collection of exit information for the defunct child process associated with the running parent process without terminating the running parent process.**

18. (Canceled)

19. (Canceled)

20. (Currently Amended) An apparatus for reaping a defunct child process associated with a **running** parent process, the apparatus comprising:

memory;

a processor coupled to memory, the processor **being** configured to:

identify a defunct child process **associated with the running parent process,** and

attach an agent thread to ~~a~~ **the running** parent process to allow modification of the **running** parent process, wherein the **running** parent process is modified to reap the defunct child process **without terminating the running parent process**.

21. (Original) The apparatus of claim 20, wherein a control criterion is used to determine whether to modify the parent process.

22. (Original) The apparatus of claim 21, wherein the control criterion comprises determining whether the parent process is stopped.

23. (Original) The apparatus of claim 21, wherein the control criterion comprises determining whether the defunct process is a newly defunct process, wherein the parent is modified only if it is determined that the defunct process is not a newly defunct process.

24. (Original) The apparatus of claim 21, wherein the control criterion comprises determining that the parent process is not an initial process.